8. Problem Set for

Advanced Algorithms II — Spring 2017

Wolfgang Mulzer, Katharina Klost

Due 12. June 2017, in class.

Problem 1 Set Cover I *10 Points*

In this problem, we will see a simple analysis of the greedy approximation algorithm for the unweighted set cover problem. Let $X = \{x_1, \ldots, x_n\}$ be a set and $S = \{S_1, \ldots, S_k\}$ a set of subsets of X. Recall that the greedy algorithm iteratively picks a set that covers the largest number of uncovered elements. Let OPT be an optimal set cover for X.

(a) Let n_j, $j \geq 0$, be the number of uncovered elements after the greedy algorithm has chosen j sets. Show that the $(j+1)$th set will cover at least $n_j/|\text{OPT}|$ new elements in X.

(b) Show that for $j \geq 0$, we have $n_{j+1} \leq (1 - 1/|\text{OPT}|)n_j$.

(c) Conclude that the greedy algorithm uses at most $|\text{OPT}| \ln n$ sets.

Problem 2 Set Cover II *10 Points*

Let $X = \{x_1, \ldots, x_n\}$ be a set and $S = \{S_1, \ldots, S_k\}$ a set of subsets of X. Furthermore, let $w : S \to \mathbb{R}^+$ be a weight function. The *weight* of a set cover $S' \subseteq S$ is $w(S') = \sum_{S_j \in S'} w(S_j)$. The goal is to find a set cover of minimum weight.

(a) Let $\mathcal{A}$ be the greedy approximation algorithm for SetCover from class, and let $m = \max\{|S_i| \mid S_i \in S\}$. Show that $\mathcal{A}$ computes an H_m-approximation for the weighted set cover problem.

(b) For any given n and $\varepsilon > 0$, construct an input X, S, w for SetCover with $|X| = n$, so that $\mathcal{A}$ will find a set cover with weight exactly H_n, even though there is a set cover of cost at most $1 + \varepsilon$. Whenever the algorithm has a choice between different sets, you may assume that it makes the worst possible choice. How does your result relate to part (a)?

Problem 3 Vertex Cover *10 Points*

(a) Let $T = (V, E)$ be a tree. Give a greedy algorithm that finds a minimum vertex cover for T in $O(|V|)$ time.

(b) Let $\mathcal{A}$ the greedy approximation algorithm for VertexCover from class. For any given n, construct a graph G with n nodes such that $\mathcal{A}$ applied to G returns a vertex cover with $2|\text{OPT}|$ nodes. You may assume that the algorithm always makes the worst possible choice.